FIG. 22 is a flowchart illustrating the top-level operation of the "Configure Graphics Node Configuration Files" step illustrated in FIG. 19.

FIG. 23 is a diagram illustrating example configuration files and screens.

FIG. 24 is a diagram illustrating example configuration files and screens.

FIG. 25 is a diagram illustrating certain slave configurations.

FIG. 26 is a diagram illustrating a system configuration for a 1x3 display.

FIG. 27 is a diagram illustrating a system configuration for a 2x2 display.

FIG. 28 is a diagram illustrating a three-tiered system configuration.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In general, the present invention is broadly directed to a system for effectively and efficiently configuring a plurality of computers to cooperate to collectively render a single graphic display, where each computer processes and renders the graphics of a portion of the display. It has been found that, in systems of this type, configuring the various computers can be a cumbersome and problematic process. Each such computer is generally equipped with a graphics card that contains the hardware and other processing logic for processing and rendering graphics to a display, and such graphics cards are typically designed to be highly configurable. Depending upon the particular configuration of the graphics cards among the various computers, the graphics cards may be in any of a number of operating states. For a plurality of computers to cooperatively render a single display, it is important that their respective graphics cards be operating in compatible states. For example, if a graphics card on a first computer is configured to run in "stereo" mode while a graphics card on a second computer is configured to run in "mono" mode, the two graphics cards would not be able to cooperate properly to render a

display. Therefore, it is important that all such graphics cards be configured to operate in compatible (although not necessarily identical) states or modes.

One way of configuring such compatible operation is to separately and independently configure each individual computer's graphics card. The manner in which graphics cards are initialized and configured is known by persons skilled in the art, and need not be described herein. As is known, however, at a higher level certain configuration options or commands may be specified through a configuration file that is stored under a known name an in a known location. These options or commands can specify the operating conditions of the display, such as the display resolution, mode, etc. Again, the specific manner in which such configuration files are processed to initialize and configure a graphics card is known and need not be described herein.

Therefore, it is important that the various configuration files (or other mechanism utilized to configure the individual graphics cards) are consistently or compatibly scripted. If even a single graphics card in such a system is incompatibly configured with the graphics cards of the remainder of the computers, then the collection of computers may not be able to render the display. It should be appreciated that, as the number of cooperating computers increases, the probability of error in consistently configuring all of the computers increases. In addition, to the extent that each of the computers is redundantly configured, as the number of computers increases, the amount of duplicative configuration effort increases.

The present invention addresses this these shortcomings by providing an effective and efficient system and method for consistently configuring a plurality of computers to cooperate to render a graphics display. As will be discussed in more detail below, the preferred inventive system and method operates by translating graphics configuration information that may be provided in a single configuration file into configuration

information suitable for communication to a plurality of computers that cooperated to render a single display. This information may be separately communicated to the various computers in the plurality of computers by way of separate files (stored in predetermined locations) or by way of direct communication through a communication port or socket.

**Illustrative Environment of the Present Invention**

Before describing the preferred embodiment of the present invention, an illustrative environment will first be described. In this regard, the present invention may be used to configure a multi-processor/multi-pipeline graphics system for rendering graphics on a single graphics display. FIG. 3 depicts a computer graphical display system 50 in accordance with such a preferred environment. As shown by FIG. 3, the system 50 includes a client 52, a master graphics pipeline 55, and one or more slave graphics pipelines 56-59. The client 52 and pipelines 55-59 may be implemented via hardware, software or any combination thereof. It should be noted that the embodiment shown by FIG. 3 depicts four slave pipelines 56-59 for illustrative purposes only, and any number of slave pipelines 56-59 may be employed to implement the system in other embodiments. As shown by FIG. 3, the pipelines 55-59, frame buffers 65-69, and compositor 76 that render graphical data to a single display device 83 are collectively referred to herein as a graphical accelerations unit 95.

The master pipeline 55 receives graphical data from the application 17 stored in the client 52. The master pipeline 55 preferably renders two-dimensional (2D) graphical data to frame buffer 65 and routes three-dimensional (3D) graphical data to slave pipelines 56-59, which render the 3D graphical data to frame buffers 66-69, respectively. Except as otherwise described herein, the client 52 and the pipelines 55-59 may be configured similar to pipelines described in U.S. patent application serial number